CLAIMS

1. A method for taking over a failed filer owning disks that store file service data and owning at least one disk that is free of file service data by a clustered partner filer, the failed filer being adapted to perform a coredump in which, in the event of failure, memory contents of the failed filer are transferred to a disk, the method comprising the steps of:

changing, by the failed filer, a coredump attribute on the disk that is free of file service data (the "coredump disk") from a non-coredump state to a coredump state and maintaining the coredump attribute on other disks owned by the failed filer in a non-coredump state;

writing the memory contents to the coredump disk;

identifying, by the clustered partner filer, the coredump attribute of the other disks and taking ownership of the other disks while allowing the failed filer to maintain ownership of the coredump disk;

upon completion of the writing of the memory contents, changing the coredump attribute to a non-coredump state; and

upon identification of the non-coredump state in the coredump attribute of the coredump disk, taking ownership, by the clustered partner filer, of the coredump disk.

2. The method as set forth in claim 1 where the coredump disk is a spare disk owned by the failed filer.

3. The method as set forth in claim 1 where the coredump disk is a disk dedicated to a coredump.

4. The method as set forth in claim 1 further comprising the step of creating, from the memory contents written on the coredump disk, a coredump data set for diagnosis of the failing filer and writing the coredump file to the file system root of the failing filer.

5. The method as set forth in claim 1 where the step of writing the memory contents includes limiting writing to the coredump disk to a predetermined time limit following a failure event of the failed filer.

6. The method as set forth in claim 5 wherein the predetermined time limit is less than a maximum time a panic of a network communicating with each of the failed filer and the clustered partner filer occurs.

7. The method as set forth in claim 5 wherein the step of limiting includes, upon elapsing of the predetermined time limit, changing the coredump attribute of the coredump disk to a non-coredump state.

8. The method as set forth in claim 1 wherein the coredump disk and each of the other disks owned by the failed filer include a plurality of predetermined regions including a coredump region and a file system data storage region and wherein the step of writing of the memory contents includes writing the memory contents into the file system data storage region of the coredump disk.

9. The method as set forth in claim 8 wherein the coredump region includes a coredump header adapted to store the coredump attribute.

10. A storage system including a first server owning interconnected first storage devices and a second server owning interconnected second storage devices, the first server and the second server being connected together by a cluster interconnect so that the second server can take over ownership of the first storage devices upon failure of the first server; the storage system comprising:

a coredump function that (a) causes the first server to write its memory to a coredump storage device chosen from one of the first storage devices in response to a sensed failure of the first server, each of the first storage devices including a coredump attribute (b) causes the attribute of the coredump storage device to be set to a coredump

state and the coredump attribute of other of the first storage devices to be set to a non-coredump state; and

a takeover function that (a) identifies each of the first storage devices with the coredump attribute set to the non-coredump state, (b) changes of each of the second devices having the coredump attribute set to the non-coredump state from ownership by the first server to ownership by the second server so that takeover of the ownership can proceed in parallel with the writing of the memory to the coredump storage device.

11. The storage system as set forth in claim 10 wherein the takeover function is adapted to cause the second server to logically assume ownership of the coredump storage device after the earlier of either an elapsing of a predetermined time limit or a completion of the writing of the memory to the coredump storage device.

12. The storage system as set forth in claim 11 wherein the coredump function is adapted to cause the coredump attribute of the coredump storage device to be changed to a non-coredump state after either the elapsing of the predetermined time limit or the completion of the writing of the memory to the coredump storage device.

13. The storage system as set forth in claim 12 wherein the non-coredump state comprises each of an aborted state, a completed state and a non-active state.

14. The storage system as set forth in claim 10 wherein the first storage devices each comprise a disk drive engaged in file service activity and the coredump storage device comprises a spare disk that is free of file service activity.

15. A computer-readable medium in a storage system that includes a first server owning interconnected first storage devices and a second server owning interconnected second storage devices, the first server and the second server being connected together by a cluster interconnect so that the second server can take over ownership of the first stor-

age devices upon failure of the first server; the computer-readable medium including program instructions for performing the steps of:

writing, by the first server, its memory to a coredump storage device chosen from one of the first storage devices in response to a sensed failure of the first server, each of the first storage devices including a coredump attribute;

setting the attribute of the coredump storage device to be set to a coredump state and the coredump attribute of other of the first storage devices to be set to set to a non-coredump state;

identifying each of the first storage devices with the coredump attribute set to the non-coredump state;

changing of each of the second devices having the coredump attribute set to the non-coredump state from ownership by the first server to ownership by the second server so that takeover of the ownership can proceed in parallel with the writing of the memory to the coredump storage device.

16. The computer-readable medium as set forth in claim 15 further comprising the step of causing the second server to logically assume ownership of the coredump storage device after the earlier of either an elapsing of a predetermined time limit or a completion of the writing of the memory to the coredump storage device.

17. The computer-readable medium as set forth in claim 16 further comprising the step of causing the coredump attribute of the coredump storage device to be changed to a non-coredump state after either the elapsing of the predetermined time limit or the completion of the writing of the memory to the coredump storage device.

18. The computer-readable medium as set forth in claim 17 wherein the non-coredump state comprises each of an aborted state, a completed state and a non-active state.

19. The computer-readable medium as set forth in claim 18 wherein the first storage devices each comprise a disk drive engaged in file service activity and the coredump storage device comprises a spare disk that is free of file service activity.

20. The computer-readable medium as set forth in claim 15 wherein each of the first storage devices includes a coredump information region and a file system region and wherein the memory is written into the file system region of the coredump storage device.

21. The computer-readable medium as set forth in claim 15 further comprising the step of creating, with the second server, a coredump data set from the memory written to the coredump storage device, the data set being adapted to enable diagnosis of a fault relative to the first server.

22. The computer-readable medium as set forth in claim 21 further comprising the step of writing the coredump data set by the second server to a file system root of the first server stored on the first storage devices.

23. A method for takeover in a storage system that includes a first server owning interconnected first storage devices and a second server owning interconnected second storage devices, the first server and the second server being connected together by a communication interconnect so that the second server can take over ownership of the first storage devices upon failure of the first server, the method comprising the steps of:

writing, by the first server, its memory to a coredump storage device chosen from one of the first storage devices in response to a sensed failure of the first server, each of the first storage devices including a coredump attribute;

setting the attribute of the coredump storage device to be set to a coredump state and the coredump attribute of other of the first storage devices to be set to set to a non-coredump state;

identifying each of the first storage devices with the coredump attribute set to the non-coredump state; and

changing each of the second storage devices having the coredump attribute set to the non-coredump state from ownership by the first server to ownership by the second server so that takeover of the ownership can proceed in parallel with the writing of the memory to the coredump storage device.

24. The method as set forth in claim 23 further comprising the step of causing the second server to logically assume ownership of the coredump storage device after the earlier of either an elapsing of a predetermined time limit or a completion of the writing of the memory to the coredump storage device.

25. The method as et forth in claim 24 wherein the step of changing includes setting a reservation on each of second storage devices so as to establish ownership by the second server.

26. The method as set forth in claim 25 wherein the reservation comprises a SCSI-3 reservation.